Claims:

1. A method for handling packet flows, comprising sequences of data packets, in a communication or computer system, the method comprising: assigning an exit number to each said packet; queuing said packets in buffer means; and outputting the queued packets in a predetermined order according to an order list determined by said exit numbers assigned to each packet before said packet is queued.

2. A method as claimed in claim 1, wherein said exit number information is assigned to packet records, which are queued in a separate buffer means to that in which said packets are queued.

3. A method as claimed in claim 2, wherein the packet records are of fixed length.

4. A method as claimed in claim 2, wherein the packet records are shorter than said packets.

5. A method as claimed in claim 1, wherein the buffer means for said packet records comprise groups of bins, each bin containing a range of exit numbers, the bins for higher exit number packet records having a larger range than bins for lower exit number packet records.

6. A method as claimed in claim 5, wherein lower exit number packet records in a bin are subdivided into a plurality of bins, each containing packet records corresponding to a smaller range of exit numbers.

7. A method as claimed in claim 5, wherein under circumstances in which a packet record is assigned an exit number corresponding to a bin that is currently being emptied, that packet record is held in a secondary bin of the same exit number for emptying after the said bin that is currently being emptied.

BEST AVAILABLE COPY

8. A method as claimed in claim 6, wherein the bins are FIFO buffers.

9. A method as claimed in claim 6, wherein the bins are LIFO stacks.

10. A method as claimed in claim 6, wherein the bins are a mixture of FIFO buffers and LIFO stacks.

11. A method as claimed in claim 5, wherein queue management is performed by (a) processing all of said bins in parallel and (b) inserting incoming data into a bin by means of a parallel processor.

12. A method as claimed in claim 11, wherein said parallel processor performing said inserting step (b) is an array processor.

13. A method as clamed in claim 12, wherein said array processor performing said inserting step (b) is a SIMD processor.

14. A communication network processor in which traffic flows consist of data packet streams, the network processor comprising: means for assigning a buffer exit number to each said packet; buffer means for queuing said packets; and output means for outputting the queued packets in a predetermined order according to an order list determined by said exit numbers assigned to each packet before said packet is queued.

15. Means for sorting data in a computer system in which data to be processed consist of data packet streams, comprising: means for assigning a buffer exit number to each said packet; buffer means for queuing said packets; and output means for outputting the queued packets in a predetermined order according to an order list determined by said exit numbers assigned to each packet before said packet is queued.

16. A parallel processor adapted to implement the method according to any of claims 1 to 13.

BEST AVAILABLE COPY

17. A parallel processor using SIMD and adapted to implement the method according to any of claims 1 to 13.

18. A network processor adapted to perform data packet management in a communication network as claimed in claim 14.

19. A processor adapted to perform data packet sorting in a computer system as claimed in claim 15.

BEST AVAILABLE COPY